

## 4. [Java] ??? ??? - return new String(arr);

<https://school.programmers.co.kr/learn/courses/30/lessons/120895>

?? ??

```
class Solution {
    public String solution(String my_string, int num1, int num2) {
        String[] arr = String.toCharArray(my_string);
        char tmp = arr[num1];
        arr[num1] = arr[num2];
        arr[num2] = tmp;
        return arr.toString();
    }
}
```

? String[] arr = String.toCharArray(my\_string);

- → String.toCharArray()는 char[]를 반환하는 메서드. String[]를 char[]로 변환하여 반환한다.

? String.toCharArray() ??

- → my\_string.toCharArray();
- toCharArray()는 String 객체를 char[]로 변환한다.
- String.toCharArray()는 (String str, int start, int end)로 호출된다.
- String 객체 (my\_string)의 start부터 end까지의 문자를 char[]로 반환한다.

? return arr.toString();

- char[]의 toString()은 String 객체를 반환한다.
- → new String(arr)은 char[]를 String 객체로 변환한다.

?? ??

```
class Solution {
    public String solution(String my_string, int num1, int num2) {
        char[] arr = my_string.toCharArray();
        char tmp = arr[num1];
        arr[num1] = arr[num2];
        arr[num2] = tmp;
        return new String(arr);
    }
}
```

??? VS ????

1. 在 `main` 函数中，定义一个指向 `int` 类型的指针变量 `ptr`，并初始化它，使其指向堆内存。

```

// static method to parse a string into an integer
// Example: Integer.parseInt("123") returns 123
// Note: This method throws a NumberFormatException if the string is not a valid integer.

```

我们使用 `new` 来创建字符串对象。在 JavaScript 中，字符串是 `String` 类型的。我们使用 `str.length()` 来获取字符串的长度。我们使用 `str` 来访问字符串。我们使用 `str` 来访问字符串。我们使用 `str` 来访问字符串。

```

// 调用 person1 的 sayHello() 方法
person1.sayHello()

// 调用 person2 的 sayHello() 方法
person2.sayHello()

// 输出结果
person1 的 sayHello() 方法
person2 的 sayHello() 方法

```

0000 0000 0000 000 000 0000 000 000 0 0000 , 000 0000 00 00  
 00 000 000 0 0000 . 0000 000 000 00 0 000 0 000 00 0000 00 .  
 0000 0000 000 00000 0000 000 , 00 00 0000 00 0000 00000  
 000 0000 0000 00 00 .

? 1. ??? (Instance Method)

- `new` 키워드 없이도 객체를 생성할 수 있다
- `new` 키워드 없이도 객체를 생성할 수 있다

예제

- `new` 키워드 없이도 객체를 생성할 수 있다
- `new` 키워드 없이도 객체를 생성할 수 있다

예제

```
public class Person {
    String name;

    public void sayHello() {
        System.out.println("Hello, my name is " + name);
    }
}

// 실행
Person p = new Person();
p.name = "Dain";
p.sayHello(); // 출력: Hello, my name is Dain
```

## ? 2. ??? ??? (Class Method)

예제

- `static` 키워드 없이도 메서드를 호출할 수 있다
- `static` 키워드 없이도 메서드를 호출할 수 있다

예제

- `static` 키워드 없이도 메서드를 호출할 수 있다
- `static` 키워드 없이도 메서드를 호출할 수 있다
- `static` 키워드 없이도 메서드를 호출할 수 있다

예제

```
public class MathUtil {
    public static int add(int a, int b) {
        return a + b;
    }
}
```

```
}

// 計算
int result = MathUtil.add(3, 5); // 3 + 5 = 8
```

---

Revision #16

Created 19 May 2025 00:40:16 by Dain

Updated 23 May 2025 04:27:47 by Dain